Automatic Topic Hierarchy Generation Using WordNet

Gerhard Brey (dh@brey.org.uk), Miguel Vieira (jose.m.vieira@kcl.ac.uk)

June 22, 2012

In order to make full use of the rich content of large text collections various finding aids are needed. One very effective way of accessing this kind of collection is via a subject taxonomy or a topic hierarchy. Most subject classification techniques [Sebastiani 2002] are based on supervised methods and need a substantial amount of training data that are used by the various machine learning algorithms on which they are based. In many cases this constitutes a significant problem if the resources to create these training data are not available.

Unsupervised methods such as clustering algorithms, though not requiring the same resources in data preparation as machine-learning based methods, need considerable attention after the techniques have been applied in order to make the clusters meaningful to users. The use of existing powerful tools such as the semantic tagger developed at Lancaster University[1] avoids these problems, providing semantic tags for each document, but often these semantic tags are very general and therefore not ideal for a user who searches for more concrete subject terms.

The aim of the research described here is the automatic generation of a topic hierarchy, using WordNet [Miller 1995, Fellbaum 1998] as the basis for a faceted browse interface, with a collection of 19th-century periodical texts as the test corpus.

Our research was motivated by the Castanet algorithm, a technique developed by Marti Hearst and Emilia Stoica [Stoica 2004, Stoica 2007] to automatically generate metadata topic hierarchies. Castanet was developed and successfully applied to short descriptions of documents. In our research we attempt to adapt and extend the Castanet algorithm so that it can be applied to the text of the actual documents for the many collections for which no abstracts or summaries are available. It should also be a viable alternative to the other techniques mentioned above.

[1] http://ucrel.lancs.ac.uk/usas/

Methodology

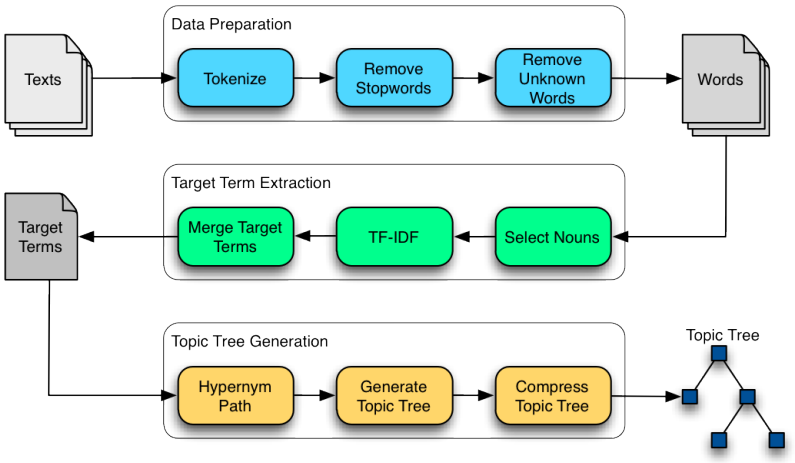

Figure 1: Algorithm workflow

The algorithm for the automatic generation of the topic hierarchy is implemented using Python with the NLTK,[2] Networkx[3] and PyGraphviz[4] modules. It has three main processes:

1. Data preparation: data needs to be prepared so that the information contained within the texts is more easily accessible [Pyle 1999].

2. Target term extraction: select terms that are considered relevant to classify each text.

3. Topic tree generation: build the tree using the target terms.

Data Preparation

The first process in the algorithm, data preparation, is also the most important in any text mining application. Data preparation leads to understanding the data and ensures that the processes that follow will be able to get the most out of the data. The data preparation process has three main steps:

1. Tokenise each of the texts in the corpus.

2. Remove stop words from the list of tokens.

[2] http://www.nltk.org/
[3] http://networkx.lanl.gov/
[4] http://networkx.lanl.gov/pygraphviz/

3. Remove unknown tokens from the list by using regular expressions and by doing lookups in WordNet's lemma dictionary. This also has the beneficial effect of removing words that are badly OCRed. Because WordNet is the basis for our topic hierarchy we regard words that not appear in its lexical database as irrelevant.

Target Term Extraction

Of all the words left in the texts after the data preparation, the algorithm only uses a subset of the most relevant terms to create the topic tree. To select the target terms:

1. Select only the ones that are nouns by performing lookups in WordNet.
2. Compute TF.IDF (Term Frequency x Inverse Document Frequency) [Manning 1999] for each one of the nouns.

   There are many ways to compute term relevance. Initially we tested the algorithms described in Castanet, information gain [Mitchell 1997] and term distribution [Sanderson 1999], but they did not produce good results for our texts as the target terms were not meaningful enough. Therefore, we decided to use TF.IDF to select the target terms as this was a more straightforward approach and the results we were getting were more relevant.
3. Select the fifteen highest scoring terms and merge them all into a unique list of target terms.

Topic Tree Generation

The list of target terms constitutes the input for the topic tree generation process:

1. Using WordNet generate a hypernym path for each of the target terms.
2. Using the hypernym paths construct a tree by joining all the paths.
3. And finally, because the hypernym path lengths are varied and to make the tree more usable/readable the tree is compressed. To compress the tree:

   (a) Starting from the leaves, recursively eliminate a parent that has less than two children, unless the parent is the root node.

   (b) Eliminate child nodes whose name appears within the parent's node.

   (c) Eliminate selected top-level nodes, because they denote very general categories and have a very broad meaning.

Examples

Consider the following hypernym paths for the nouns:

- writings: communication.n.02, written_communication.n.01, writing.n.02, sacred_text.n.01, hagiographa.n.01.
- charter: communication.n.02, written_communication.n.01, writing.n.02, document.n.01, charter.n.01.
- criticism: communication.n.02, message.n.02, disapproval.n.02, criticism.n.01.

This following image shows the previous hypernym paths combined to create a topic tree. There are many parents with just one child and the height of the tree varies across its different paths.

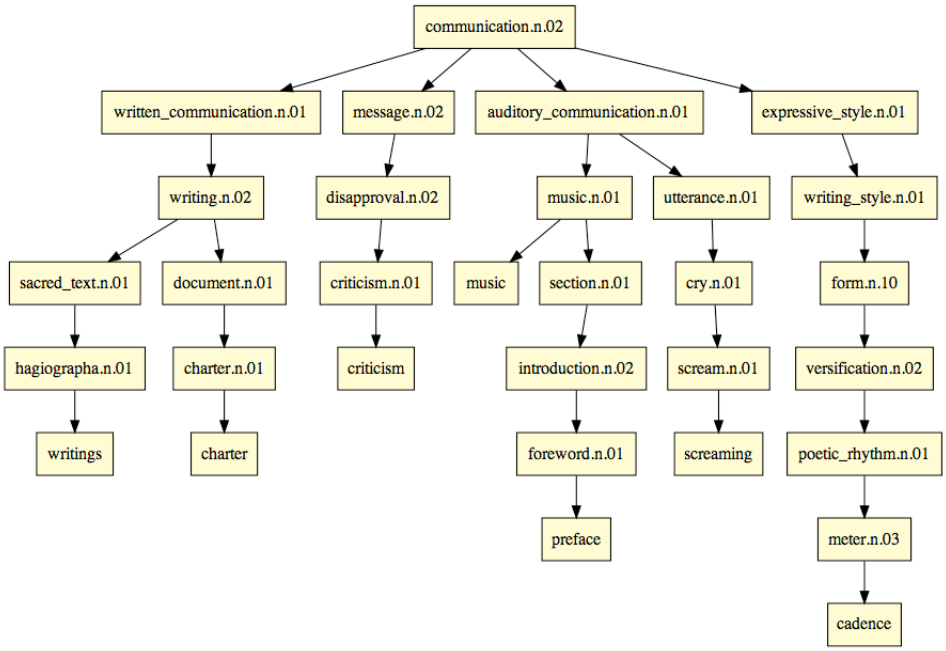

Figure 2: Communication full tree

In the following image we can see how the tree compression works. The grey nodes will be removed because they have fewer than two children, and the black nodes will be removed because their names appear within the parent node.

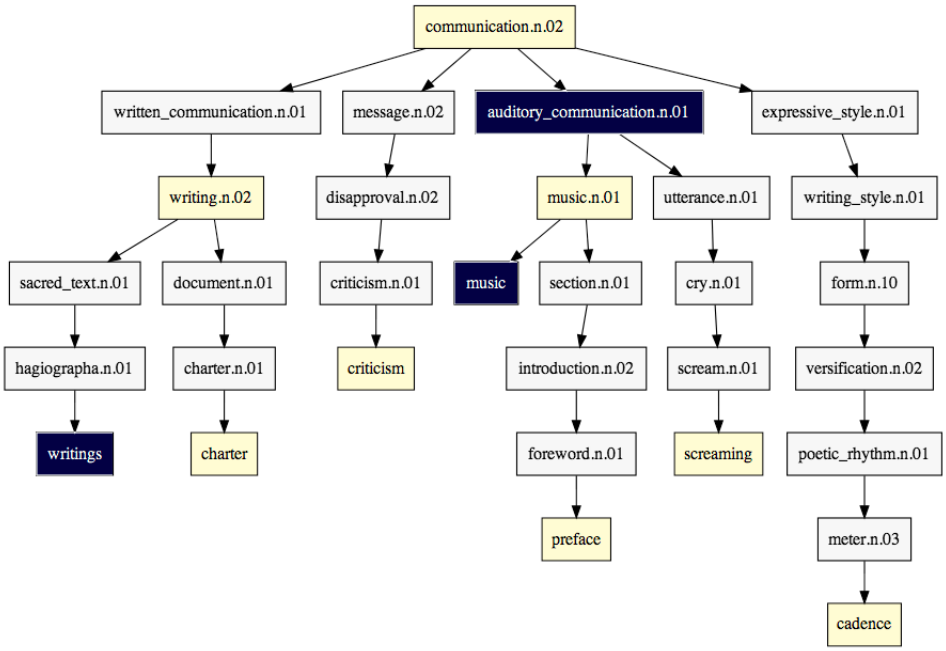

Figure 3: Topic tree compression

After the grey and black nodes are removed we get the compressed version of the tree. This tree is not as deep, therefore making it easier to navigate and use from a faceted browsing point of view.

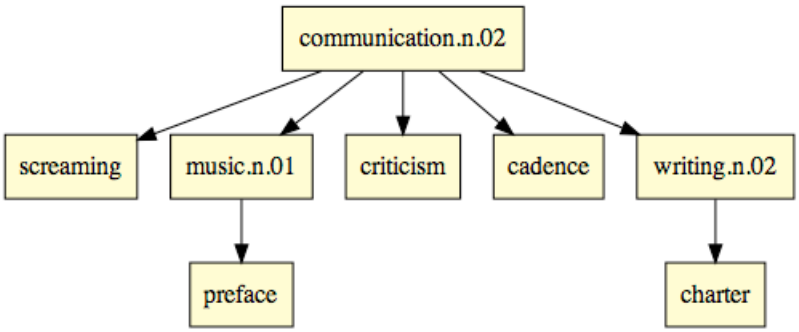

Figure 4: Communication compressed tree

Results and Future Work

The text collection used to test the techniques tried in our research is a digital edition of a 19th century periodical, the English Women's Journal (EWJ). The

EWJ is one subcollection of the Nineteenth Century Serials Edition (NCSE),[5] a digital edition of 6 newspapers and periodicals from the nineteenth century. Although the smallest collection within NCSE it was still large and varied enough to serve as a test corpus for our research. EWJ was published monthly between 1858 and 1864. It was edited and written by women and treated mainly literary, political and social contents. The collection is made up of 78 issues containing a total of 7964 articles. The main reason to choose the EWJ collection as our test corpus was that of the 6 periodicals contained in NCSE it is the one with the best OCR quality. Nevertheless the OCR quality of the EWJ was still a problem we had to contend with.

Our test corpus is composed of 1359 texts, with a minimum of 300 characters each, containing about 3.5 million words in total. The list of target terms has 8013 unique terms, when selecting the top fifteen target terms per text. The resulting topic tree has 18234 nodes and after compression it is roughly 50% smaller.

Of the samples we evaluated, over 90% of the topics are relevant, i.e. they clearly illustrate what the articles are about and the topic hierarchy adequately relates to the content of the articles. The results were evaluated by selecting a sample of 20 articles and generating the topic hierarchy for them. After reading each of these articles we compared its contents with the terms in the topic hierarchy, and counted the relevant terms for each one of them.

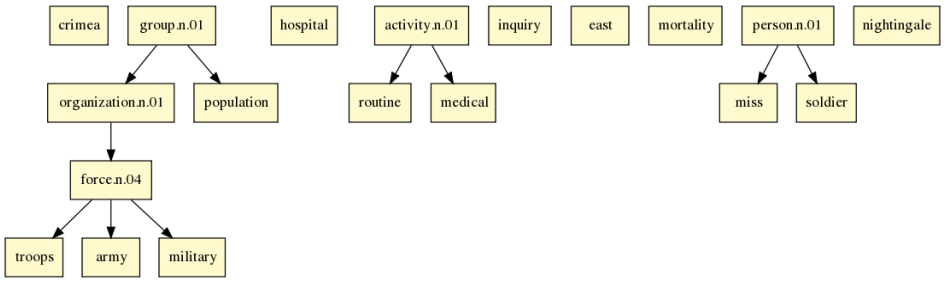

Figure 5: Topic hierarchy for an article about Florence Nightingale and the Crimean war [EWJ 1858]

Even though we consider that the algorithm produces good and promising results, we identified problems that mainly relate to the nature of our corpus. Despite our best efforts to filter out mis-OCRed portions of text, it was unavoidable that some misleading tokens remained. This sometimes led to strange results in the topic tree. For example, the partial text "Another act in the great European drama" from the original article was OCRed into "Ano , ther act in the great European drama". The word "Ano", which was not filtered out in the data preparation process, is classified in WordNet as the Abu Nidal Organization, which did not exist in the 19th century.

[5] http://www.ncse.ac.uk/

Another source of problems was the erroneous disambiguation of tokens with multiple meanings in WordNet. An example of this is the word "drama", from the sentence above. It was output in the topic hierarchy as referring to a theatrical play rather than in its figurative sense as a turbulent event.

Future work could explore better ways how to deal with bad OCR. We also plan to enhance topic disambiguation; this could be achieved by analysing the domains of the topics in the hypernym paths before deciding which one to add to the topic tree.

Faceted browsing interfaces based on topic hierarchies are easy and intuitive to navigate [Morville 2006], and as our results demonstrate, topic hierarchies as generated by our approach form an appropriate basis for this type of data navigation. We are confident that our approach can successfully be applied to other corpora and should yield even better results if there are no OCR issues to contend with. And since WordNet is available in several languages,[6] it should also be possible to apply our approach to corpora in other languages.

References

[Sebastiani 2002] Fabrizio Sebastiani. 2002. "Machine Learning in Automated Text Categorization". ACM Computing Surveys, 34.1, 1-47.

[Miller 1995] George A. Miller (1995). WordNet: A Lexical Database for English. Communications of the ACM Vol. 38, No. 11: 39-41.

[Fellbaum 1998] Christiane Fellbaum (1998, ed.) WordNet: An Electronic Lexical Database. Cambridge, MA: MIT Press.

[Stoica 2004] Emilia Stoica and Marti A. Hearst. 2004. "Nearly-automated metadata hierarchy creation". Proceedings of HLT-NAACL 2004: Short Papers, Boston, Mass., 117-120.

[Stoica 2007] Emilia Stoica, Marti A. Hearst and Megan Richardson. 2007. "Automating Creation of Hierarchical Faceted Metadata Structures"'. Proceedings of NAACL/HLT 2007, Rochester, NY, April, 244-251.

[Pyle 1999] Doryan Pyle. 1999. Data Preparation for Data Mining. San Francisco: Morgan Kaufman Publishers Inc, chapter 3.

[Manning 1999] Term frequency x inverse document frequency. See: Christopher D. Manning, Prabhakar Raghavan and Hinrich Schütze. 1999. Introduction to Information Retrieval. Cambridge: CUP, p. 116-123.

[Mitchell 1997] Tom Mitchell. 1997. Machine Learning. New York: McGraw Hill, pp. 57-60.

[6] http://en.wikipedia.org/wiki/WordNet#Other_languages

[Sanderson 1999] Mark Sanderson and Bruce Croft. 1999. "Deriving concept hierarchies from text". Proceedings of the 22nd annual international ACM SIGIR conference on Research and Development in Information Retrieval 1999, 206-213.

[EWJ 1858] English Women's Journal, 1 April 1858, pp. 73-79.

[Morville 2006] Peter Morville and Louis Rosenfeld. 2006. Information Architecture for the World Wide Web, Sebastopol: O'Reilly, pp. 221-227.